

1. A method comprising:  
receiving input data by an execution unit; and  
performing by the execution unit one or more current multiply-accumulate operations on  
the received input data and on input data received by the execution unit for one or more prior  
5 multiply-accumulate operations and saved by the execution unit.

2. The method of claim 1, wherein the receiving comprises receiving first and second data  
by the execution unit; and  
wherein the performing comprises performing by the execution unit a multiply-  
10 accumulate operation on the received first and second data and a multiply-accumulate operation  
on the received first data and on input data saved by the execution unit.

3. The method of claim 1, wherein the receiving comprises receiving first, second, third,  
and fourth data by the execution unit; and  
15 wherein the performing comprises performing by the execution unit a multiply-  
accumulate operation on the received first and third data, a multiply-accumulate operation on the  
received second and fourth data, a multiply-accumulate operation on the received first and fourth  
data, and a multiply-accumulate operation on the received second data and on input data saved  
by the execution unit.

20 4. The method of claim 3, wherein the performing the multiply-accumulate operation on the  
received first and third data and the multiply-accumulate operation on the received second and  
fourth data comprise multiplying the received first and third data to produce a first product,

multiplying the received second and fourth data to produce a second product, and adding the first product, the second product, and an accumulated sum.

5. The method of claim 1, comprising:

5 saving by the execution unit received input data for one or more multiply-accumulate operations to be performed by the execution unit after the current multiply-accumulate operations;

repeating the receiving, performing, and saving by the execution unit one or more times to accumulate data; and

10 outputting the accumulated data by the execution unit.

6. The method of claim 5, comprising performing the repeating to implement a finite impulse response filter, wherein the received input data comprises one or more tap coefficients and one or more input data samples and wherein the accumulated data comprises one or more

15 output data samples.

7. The method of claim 6, wherein one or more tap coefficients are each a complex number and one or more input data samples are each a complex number.

20 8. The method of claim 1, comprising saving by the execution unit saved input data for one or more multiply-accumulate operations to be performed by the execution unit.

9. The method of claim 1, comprising performing the receiving and performing in accordance with a single instruction multiple data instruction.

11/14/2009 10:00:00 AM

10. An apparatus comprising:

an execution unit block having a plurality of inputs, the execution unit block comprising one or more buffers to save input data received at one or more of the inputs and a plurality of multiplier-accumulators to perform multiply-accumulate operations, the multiplier-accumulators to perform current multiply-accumulate operations on input data received at one or more of the inputs and on input data received by the execution unit block for one or more prior multiply-accumulate operations and stored in one or more buffers of the execution unit block; and control logic to control the multiplier-accumulators and the one or more buffers.

11. The apparatus of claim 10, the multiplier-accumulators to perform a multiply-accumulate operation on first and second input data received at the inputs and a multiply-accumulate operation on the received first input data and on input data stored in a buffer of the execution unit block.

12. The apparatus of claim 11, wherein the execution unit block comprises a single multiplier-accumulator comprising a multiplier to multiply the received first and second input data to produce a product, an accumulator to store an accumulated sum, and an adder to add the product to the accumulated sum.

13. The apparatus of claim 10, the multiplier-accumulators to perform a multiply-accumulate operation on first and third input data received at the inputs, a multiply-accumulate operation on second and fourth data received at the inputs, a multiply-accumulate operation on the received

first and fourth data, and a multiply-accumulate operation on the received second data and on input data stored in a buffer of the execution unit block.

14. The apparatus of claim 13, wherein the execution unit block comprises a dual multiplier-accumulator comprising a multiplier to multiply the received first and third input data to produce a first product, a multiplier to multiply the received second and fourth input data to produce a second product, an accumulator to store an accumulated sum, and an adder to add the first product, the second product, and the accumulated sum.

15. The apparatus of claim 10, the execution unit block comprising one or more execution unit building blocks each comprising one or more multiplier-accumulators and one or more buffers.

16. The apparatus of claim 10, the control logic to control the execution unit block to repeat, one or more times, receiving input data at one or more of the inputs, performing multiply-accumulate operations on the received input data and on input data stored in one or more buffers of the execution unit block to accumulate data, and saving the received input data in one or more buffers of the execution unit block.

17. The apparatus of claim 16, the multiplier-accumulators to perform the repeating to implement a finite impulse response filter, wherein the received input data comprises one or more tap coefficients and one or more input data samples and wherein the accumulated data comprises one or more output data samples.

18. The apparatus of claim 17, wherein one or more tap coefficients are each a complex number and one or more input data samples are each a complex number.

5 19. The apparatus of claim 10, the control logic to control the execution unit block to save input data stored in one or more buffers of the execution unit block for one or more multiply-accumulate operations to be performed by the execution unit block.

10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100

20. The apparatus of claim 10, the control logic to control the multiplier-accumulators and the one or more buffers in accordance with a single instruction multiple data instruction.

21. A system comprising:

(a) a coder/decoder to receive analog signals and convert the analog signals into corresponding input data; and

(b) a processor to process the corresponding input data, the processor comprising:

5 (i) an execution unit block having a plurality of inputs, the execution unit block comprising one or more buffers to save input data received at one or more of the inputs and a plurality of multiplier-accumulators to perform multiply-accumulate operations, the multiplier-accumulators to perform current multiply-accumulate operations on input data received at one or more of the inputs and on input data received by the execution unit block for one or more prior multiply-accumulate operations and stored in one or more buffers of the execution unit block, and

10 (ii) control logic to control the multiplier-accumulators and the one or more buffers.

15 22. The system of claim 21, the multiplier-accumulators to perform a multiply-accumulate operation on first and second input data received at the inputs and a multiply-accumulate operation on the received first input data and on input data stored in a buffer of the execution unit block.

20 23. The system of claim 22, wherein the execution unit block comprises a single multiplier-accumulator comprising a multiplier to multiply the received first and second input data to produce a product, an accumulator to store an accumulated sum, and an adder to add the product to the accumulated sum.

24. The system of claim 21, the multiplier-accumulators to perform a multiply-accumulate operation on first and third input data received at the inputs, a multiply-accumulate operation on second and fourth data received at the inputs, a multiply-accumulate operation on the received first and fourth data, and a multiply-accumulate operation on the received second data and on input data stored in a buffer of the execution unit block.

25. The system of claim 24, wherein the execution unit block comprises a dual multiplier-accumulator comprising a multiplier to multiply the received first and third input data to produce a first product, a multiplier to multiply the received second and fourth input data to produce a second product, an accumulator to store an accumulated sum, and an adder to add the first product, the second product, and the accumulated sum.

26. The system of claim 21, the execution unit block comprising one or more execution unit building blocks each comprising one or more multiplier-accumulators and one or more buffers.

27. The system of claim 21, the control logic to control the execution unit block to repeat, one or more times, receiving input data at one or more of the inputs, performing multiply-accumulate operations on the received input data and on input data stored in one or more buffers of the execution unit block to accumulate data, and saving the received input data in one or more buffers of the execution unit block.



28. The system of claim 27, the multiplier-accumulators to perform the repeating to implement a finite impulse response filter, wherein the received input data comprises one or more tap coefficients and one or more input data samples and wherein the accumulated data comprises one or more output data samples.

5

29. The system of claim 21, the control logic to control the execution unit block to save input data stored in one or more buffers of the execution unit block for one or more multiply-accumulate operations to be performed by the execution unit block.

30. The system of claim 21, the control logic to control the multiplier-accumulators and the one or more buffers in accordance with a single instruction multiple data instruction.